4 EXPLOITING SOFTWARE DIVERSIFICATION FOR WEBASSEMBLY

4.1 Offensive Diversification: Malware evasion

Binary rewriting tools and obfuscators The landscape for tools that can modify, obfuscate, or enhance WebAssembly binaries for various has increased. For instance, BREWasm[**?**] provides a comprehensive static binary rewriting framework specifically designed for WebAssembly. Wobfuscator[**?**] takes a different approach, serving as an opportunistic obfuscator for Wasm-JS browser applications. Madvex[**?**] focuses on modifying WebAssembly binaries to evade malware detection, with its approach being limited to alterations in the code section of a WebAssembly binary. Additionally, WASMixer[**?**] obfuscates WebAssembly binaries, by including memory access encryption, control flow flattening, and the insertion of opaque predicates.

TODO The malware evasion paper

4.1.1 Objective

Test and evade the resilience of WebAssembly malware detectors mentioned in Subsection 2.1.5.

4.1.2 Approach

TODO We use wasm-mutate **TODO** How do we use it? **TODO** Controlled and uncontrolled diversification.

[0] Comp. time 2023/10/04 06:38:06

4.1.3 Results

4.2 Defensive Diversification: Speculative Side-channel protection

TODO Go around the last paper

4.2.1 Threat model

- Spectre timing cache attacks.
    - Rockiki paper on portable side channel in browsers.

4.2.2 Approach

- Use of wasm-mutate

4.2.3 Results

- Diminshing of BER